CS1N

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	368	(instruction near2 port) same (data near2 port)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:07
L2	4	(instruction near2 port) same (data near2 port) same bank$1	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 09:53
L3	3	(instruction near2 port) same (data near2 port) same bank$1 same width	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 09:54
L4	178	(instruction near2 port) and (data near2 port) and bank$1 and width	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 09:55
L5	159	(instruction near2 port) and (data near2 port) and bank$1 and width and cach$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 09:55
L6	106	(instruction near2 port) and (data near2 port) and bank$1 and (data near2 width) and cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 09:56
L7	54	(instruction near2 port) and (data near2 port) and (number$1 near5 bank$1) and (data near2 width) and cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:05

EAST Search History

L8	172	(711/131).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 09:57
L9	0	7 and 8	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:06
L10	4	1 and 8	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:01
L11	428	(711/117).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 10:01
L12	475	(711/119).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 10:01
L13	204	(711/123).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 10:01
L14	280	(711/125).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 10:01
L15	111	(711/126).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 10:01
L16	348	(711/149).CCLS.	USPAT; USOCR	OR	OFF	2007/11/19 10:01
L17	1	1 and 11	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:01
L18	2	1 and 12	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:01
L19	1	1 and 13	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:01

EAST Search History

L20	5	1 and 14	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:01
L21	0	1 and 15	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:02
L22	4	1 and 16	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:02
L23	1	(instruction near2 port) and (data near2 port) and ((bigger or larger) near10 (number$1 near5 bank$1)) and (data near2 width) and cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:05
L24	0	7 and 16	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:06
L25	0	6 and 8	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:06
L26	2	(instruction near2 port) same (data near2 port) same cache same banks	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:08

EAST Search History

L27	12455	(port same bank$1)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:08
L28	275	(multi-port same bank$1)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:08
L29	3	(multi-port same bank$1) same (width)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:09

Interference searched

EAST Search History

L27	12455	(port same bank$1)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:08
L28	275	(multi-port same bank$1)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:57
L29	3	(multi-port same bank$1) same (width)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/11/19 10:09
L30	16	((instruction near2 port) same (data near2 port)).clm.	US-PGPUB	OR	OFF	2007/11/19 10:56
L31	1	((instruction near2 port) and (data near2 port) and bank$1 and width).clm.	US-PGPUB	OR	OFF	2007/11/19 10:56
L32	1	((instruction near2 port) and (data near2 port) and bank$1 and (data near2 width) and cache).clm.	US-PGPUB	OR	OFF	2007/11/19 10:56
L33	23	((multi-port same bank$1)).clm.	US-PGPUB	OR	OFF	2007/11/19 10:57

PORTAL USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

SEARCH

THE ACM DIGITAL LIBRARY Advanced Search

Search Tips

Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Desired Results:

must have **all** of the words or phrases

instruction port, data port, cache, bank

must have **any** of the words or phrases

must have **none** of the words or phrases

Name or Affiliation:

Authored by: all any none

Edited by: all any none

Reviewed by: all any none

Only search in:*

Title Abstract Review All Information

SEARCH

*Searches will be performed on all available information, including full text where available, unless specified above.

ISBN / ISSN: Exact Expand

DOI: Exact Expand

SEARCH

Published:

By: all any none

In: all any none

Since: Month Year

Before: Month Year

As: Any type of publication

SEARCH

Conference Proceeding:

Sponsored By:

Conference Location:

Conference Year: yyyy

Classification: (CCS) Primary Only

Classified as: all any none

Subject Descriptor: all any none

Keyword Assigned: all any none

Results must have accessible:

Full Text Abstract Review

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+abstract:instruction +abstract:port, +abstract:data +abstrac

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: **instruction port data port cache bank**

Found **2** of **215,186**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 2 of 2

Relevance scale

1 Scalable cache memory design for large-scale SMT architectures

Muhamed F. Mudawar

June 2004 **Proceedings of the 3rd workshop on Memory performance issues: in conjunction with the 31st international symposium on computer architecture WMPI '04**

Publisher: ACM Press

Full text available: pdf(327.97 KB) Additional Information: full citation, abstract, references

The cache hierarchy design in existing SMT and superscalar processors is optimized for latency, but not for band-width. The size of the L1 data cache did not scale over the past decade. Instead, larger unified L2 and L3 caches were introduced. This cache hierarchy has a high overhead due to the principle of containment. It also has a complex design to maintain cache coherence across all levels. Furthermore, this cache hierarchy is not suitable for future large-scale SMT processors, which will de ...

Keywords: scalable multiported cache memory, simultaneous multithreaded architectures

2 On high-bandwidth data cache design for multi-issue processors

Jude A. Rivers, Gary S. Tyson, Edward S. Davidson, Todd M. Austin

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture MICRO 30**

Publisher: IEEE Computer Society

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Publisher Site

Highly aggressive multi-issue processor designs of the past few years and projections for the decade, require that we redesign the operation of the cache memory system. The number of instructions that must be processed (including incorrectly predicted ones) will approach 16 or more per cycle. Since memory operations account for about a third of all instructions executed, these systems will have to support multiple data references per cycle. In this paper, we explore reference stream characterist ...

Keywords: Locality-Based Interleaving, Multiporting, High-Bandwidth Data Supply, Multi-Bank Caches

PORTAL USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

SEARCH

THE ACM DIGITAL LIBRARY Advanced Search

Search Tips

Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Desired Results:

must have **all** of the words or phrases

instruction port, data port, cache

must have **any** of the words or phrases

must have **none** of the words or phrases

Name or Affiliation:

Authored by: all any none

Edited by: all any none

Reviewed by: all any none

Only search in:*

Title Abstract Review All Information

SEARCH

*Searches will be performed on all available information, including full text where available, unless specified above.

ISBN / ISSN: Exact Expand

DOI: Exact Expand

SEARCH

Published:

By: all any none

In: all any none

Since: Month Year

Before: Month Year

As: Any type of publication

Conference Proceeding:

Sponsored By:

Conference Location:

Conference Year: yyyy

SEARCH

Classification: (CCS) Primary Only

Classified as: all any none

Subject Descriptor: all any none

Keyword Assigned: all any none

Results must have accessible:

Full Text Abstract Review

PORTAL USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+abstract:instruction +abstract:port, +abstract:data +abstrac

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: **instruction port data port cache** Found **28** of **215,186**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 28 Result page: **1** 2 next

Relevance scale

1 Scalable cache memory design for large-scale SMT architectures

Muhamed F. Mudawar

June 2004 **Proceedings of the 3rd workshop on Memory performance issues: in conjunction with the 31st international symposium on computer architecture WMPI '04**

Publisher: ACM Press

Full text available: pdf(327.97 KB) Additional Information: full citation, abstract, references

The cache hierarchy design in existing SMT and superscalar processors is optimized for latency, but not for band-width. The size of the L1 data cache did not scale over the past decade. Instead, larger unified L2 and L3 caches were introduced. This cache hierarchy has a high overhead due to the principle of containment. It also has a complex design to maintain cache coherence across all levels. Furthermore, this cache hierarchy is not suitable for future large-scale SMT processors, which will de ...

Keywords: scalable multiported cache memory, simultaneous multithreaded architectures

2 Tango: a hardware-based data prefetching technique for superscalar processors

Shlomit S. Pinter, Adi Yoaz

December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture MICRO 29**

Publisher: IEEE Computer Society

Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms

We present a new hardware-based data prefetching mechanism for enhancing instruction level parallelism and improving the performance of superscalar processors. The emphasis in our scheme is on the effective utilization of slack time and hardware resources not used for the main computation. The scheme suggests a new hardware construct, the program progress graph (PPG), as a simple extension to the branch target buffer (BTB). We use the PPG for implementing a fast pre-program counter pre-PC, that ...

Keywords: LRU mechanism, SPEC92 benchmark, Tango, base line architecture, branch target buffer, hardware resources, hardware-based data prefetching technique, instruction level parallelism, instruction prefetching, memory reference instructions, parallel processing, performance, program progress graph, simulation results, slack time, superscalar processors

3 Power: A compiler approach for reducing data cache energy

W. Zhang, M. Karakoy, M. Kandemir, G. Chen

June 2003 **Proceedings of the 17th annual international conference on Supercomputing ICS '03**

Publisher: ACM Press

Full text available: pdf(299.23 KB) Additional Information: full citation, abstract, references, citings, index terms

Silicon technology advances have made it possible to pack millions of transistors --- switching at high clock speeds --- on a single chip. While these advances bring unprecedented performance to electronic products, they pose difficult power/energy consumption problems. For example, large number of transistors in dense on-chip cache memories consume significant static (leakage) power even if the cache is not used by the current computation. While previous compiler research studied code and data ...

Keywords: compiler analysis, data caches, energy optimization

4 Reducing data cache leakage energy using a compiler-based approach

Wei Zhang, Mahmut Kandemir, Mustafa Karakoy, Guangyu Chen

August 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available: pdf(750.57 KB) Additional Information: full citation, abstract, references, index terms, review

Silicon technology advances have made it possible to pack millions of transistors--- switching at high clock speeds---on a single chip. While these advances bring unprecedented performance to electronic products, they also pose difficult power/energy consumption problems. For example, large number of transistors in dense on-chip cache memories consume significant static (leakage) power even if the cache is not used by the current computation. While previous compiler research studied code and data ...

Keywords: Compiler analysis, array-intensive applications, data caches, energy optimization, pointer-intensive applications

5 On the effectiveness of prefetching and reuse in reducing L1 data cache traffic: a case study of Snort

G. Surendra, Subhasis Banerjee, S. K. Nandy

June 2004 **Proceedings of the 3rd workshop on Memory performance issues: in conjunction with the 31st international symposium on computer architecture WMPI '04**

Publisher: ACM Press

Full text available: pdf(408.69 KB) Additional Information: full citation, abstract, references, index terms

Reducing the number of data cache accesses improves performance, port efficiency, bandwidth and motivates the use of single ported caches instead of complex and expensive multi-ported ones. In this paper we consider an intrusion detection system as a target application and study the effectiveness of two techniques - (i) prefetching data from the cache into local buffers in the processor core and (ii) load Instruction Reuse (IR) - in reducing data cache traffic. The analysis is carried out using ...

Keywords: cache traffic, instruction reuse, intrusion detection, network processor, pattern matching

6 Zero cost indexing for improved processor cache performance

Tony Givargis
January 2006 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 11 Issue 1
Publisher: ACM Press
Full text available: pdf(418.19 KB) Additional Information: full citation, abstract, references, index terms

The increasing use of microprocessor cores in embedded systems as well as mobile and portable devices creates an opportunity for customizing the cache subsystem for improved performance. In traditional cache design, the index portion of the memory address bus consists of the *K* least significant bits, where *K* = $\log_2 D$ and *D* is the depth of the cache. However, in devices where the application set is known and characterized (e.g., systems that execute a ...

Keywords: Cache optimization, design exploration, index hashing

7 Compiler support for software-based cache partitioning

Frank Mueller
November 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 workshop on Languages, compilers, & tools for real-time systems LCTES '95**, Volume 30 Issue 11
Publisher: ACM Press
Full text available: pdf(886.21 KB) Additional Information: full citation, abstract, references, citings, index terms

Cache memories have become an essential part of modern processors to bridge the increasing gap between fast processors and slower main memory. Until recently, cache memories were thought to impose unpredictable execution time behavior for hard real-time systems. But recent results show that the speedup of caches can be exploited without a significant sacrifice of predictability. These results were obtained under the assumption that real-time tasks be scheduled **non-preemptively**.This paper ...

8 Prefetch injection based on hardware monitoring and object metadata

Ali-Reza Adl-Tabatabai, Richard L. Hudson, Mauricio J. Serrano, Sreenivas Subramoney
June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation PLDI '04**, Volume 39 Issue 6
Publisher: ACM Press
Full text available: pdf(288.00 KB) Additional Information: full citation, abstract, references, citings, index terms

Cache miss stalls hurt performance because of the large gap between memory and processor speeds - for example, the popular server benchmark SPEC JBB2000 spends 45% of its cycles stalled waiting for memory requests on the Itanium® 2 processor. Traversing linked data structures causes a large portion of these stalls. Prefetching for linked data structures remains a major challenge because serial data dependencies between elements in a linked data structure preclude the timely materialization ...

Keywords: cache misses, compiler optimization, garbage collection, prefetching, profile-guided optimization, virtual machines

9 Dynamic dead-instruction detection and elimination

J. Adam Butts, Guri Sohi
October 2002 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , ACM**

SIGARCH Computer Architecture News , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X, Volume 36 , 37 , 30 Issue 5 , 10 , 5

Publisher: ACM

Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, cited by

We observe a non-negligible fraction--3 to 16% in our benchmarks--of *dynamically dead instructions,* dynamic instruction instances that generate unused results. The majority of these instructions arise from static instructions that also produce useful results. We find that compiler optimization (specifically instruction scheduling) creates a significant portion of these *partially dead* static instructions. We show that most of the dynamically instructions arise from a small set of st ...

10 Power efficient cache design: Reducing data cache energy consumption via cached load/store queue

Dan Nicolaescu, Alex Veidenbaum, Alex Nicolau

August 2003 **Proceedings of the 2003 international symposium on Low power electronics and design ISLPED '03**

Publisher: ACM Press

Full text available: pdf(140.51 KB) Additional Information: full citation, abstract, references, citings, index terms

High-performance processors use a large set--associative L1 data cache with multiple ports. As clock speeds and size increase such a cache consumes a significant percentage of the total processor energy. This paper proposes a method of saving energy by reducing the number of data cache accesses. It does so by modifying the Load/Store Queue design to allow "caching" of previously accessed data values on both loads and stores after the corresponding memory access instruction has been committed. It ...

Keywords: LSQ, cache, load queue, low energy, low latency, low power, memory, store queue

11 On high-bandwidth data cache design for multi-issue processors

Jude A. Rivers, Gary S. Tyson, Edward S. Davidson, Todd M. Austin

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture MICRO 30**

Publisher: IEEE Computer Society

Full text available: pdf(1.38 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Highly aggressive multi-issue processor designs of the past few years and projections for the decade, require that we redesign the operation of the cache memory system. The number of instructions that must be processed (including incorrectly predicted ones) will approach 16 or more per cycle. Since memory operations account for about a third of all instructions executed, these systems will have to support multiple data references per cycle. In this paper, we explore reference stream characterist ...

Keywords: Locality-Based Interleaving, Multiporting, High-Bandwidth Data Supply, Multi-Bank Caches

12 Cache-Aware Scratchpad Allocation Algorithm

Manish Verma, Lars Wehmeyer, Peter Marwedel

February 2004 **Proceedings of the conference on Design, automation and test in Europe - Volume 2 DATE '04**

Publisher: IEEE Computer Society

Full text available: pdf(117.95 KB) Additional Information: full citation, abstract, citings, index terms

In the context of portable embedded systems, reducing energy is one of the prime objectives. Most high-end embedded microprocessors include onchip instruction and data caches, along with a small energy ef.cient scratchpad. Previous approaches for utilizing scratchpad did not consider caches and hence fail for the au courant architecture. In thepresented work, we use the scratchpad for storing instructions and propose a generic Cache Aware Scratchpad Allocation (CASA) algorithm. We report an aver ...

13 Access region locality for high-bandwidth processor memory system design

Sangyeun Cho, Pen-Chung Yew, Gyungho Lee

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture MICRO 32**

Publisher: IEEE Computer Society

Full text available: pdf(1.43 MB) Additional Information: full citation, abstract, references, citings, index terms

Publisher Site

This paper studies an interesting yet less explored behavior of memory access instructions, called access region locality. Unlike the traditional temporal and spatial data locality that focuses on individual memory locations and how accesses to the locations are inter-related, the access region locality concerns with each static memory instruction and its range of access locations at run time. We consider program's data, heap, and stack regions in this paper. Our experimental study ...

14 Streamlining data cache access with fast address calculation

Todd M. Austin, Dionisios N. Pnevmatikatos, Gurindar S. Sohi

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture ISCA '95**, Volume 23 Issue 2

Publisher: ACM

Full text available: pdf(1.58 MB) Additional Information: full citation, abstract, references, cited by, index terms

For many programs, especially integer codes, untolerated load instruction latencies account for a significant portion of total execution time. In this paper, we present the design and evaluation of a fast address generation mechanism capable of eliminating the delays caused by effective address calculation for many loads and stores.Our approach works by predicting early in the pipeline (part of) the effective address of a memory access and using this predicted address to speculatively access the ...

15 Session S4.1: power in memory and network processors: Increasing power efficiency of multi-core network processors through data filtering

Gokhan Memik, William H. Mangione-Smith

October 2002 **Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems CASES '02**

Publisher: ACM Press

Full text available: pdf(234.17 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose and evaluate a data filtering method to reduce the power consumption of high-end processors with multiple execution cores. Although the proposed method can be applied to a wide variety of multi-processor systems including MPPs, SMPs and any type of single-chip multiprocessor, we concentrate on Network Processors. The proposed method uses an execution unit called Data Filtering Engine that processes data with low temporal locality before it is placed on the system bus. The execution co ...

Keywords: chip multiprocessors, data locality, network processors, power reduction, remote procedure call

16 Exploiting Fine-Grained Data Parallelism with Chip Multiprocessors and Fast Barriers
Jack Sampson, Ruben Gonzalez, Jean-Francois Collard, Norman P. Jouppi, Mike Schlansker, Brad Calder
December 2006 **Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture MICRO 39**
Publisher: IEEE Computer Society
Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, index terms

We examine the ability of CMPs, due to their lower onchip communication latencies, to exploit data parallelism at inner-loop granularities similar to that commonly targeted by vector machines. Parallelizing code in this manner leads to a high frequency of barriers, and we explore the impact of different barrier mechanisms upon the efficiency of this approach. To further exploit the potential of CMPs for fine-grained data parallel tasks, we present barrier filters, a mechanism for fast barrier sy ...

17 Architecture -- memory hierarchy: Performance driven data cache prefetching in a dynamic software optimization system
Jean Christophe Beyler, Philippe Clauss
June 2007 **Proceedings of the 21st annual international conference on Supercomputing ICS '07**
Publisher: ACM Press
Full text available: pdf(391.92 KB) Additional Information: full citation, abstract, references, index terms

Software or hardware data cache prefetching is an efficient way to hide cache miss latency. However effectiveness of the issued prefetches have to be monitored in order to maximize their positive impact while minimizing their negative impact on performance. In previous proposed dynamic frameworks, the monitoring scheme is either achieved using processor performance counters or using specific hardware. In this work, we propose a prefetching strategy which does not use any specific hardware com ...

Keywords: binary instrumentation, data cache prefetching, dynamic optimization

18 Compiler Managed Dynamic Instruction Placement in a Low-Power Code Cache
Rajiv A. Ravindran, Pracheeti D. Nagarkar, Ganesh S. Dasika, Eric D. Marsman, Robert M. Senger, Scott A. Mahlke, Richard B. Brown
March 2005 **Proceedings of the international symposium on Code generation and optimization CGO '05**
Publisher: IEEE Computer Society
Full text available: pdf(631.68 KB) Additional Information: full citation, abstract, citings, index terms

Modern embedded microprocessors use low power on-chip memories called scratch-pad memories to store frequently executed instructions and data. Unlike traditional caches, scratch-pad memories lack the complex tag checking and comparison logic, thereby proving to be efficient in area and power. In this work, we focus on exploiting scratch-pad memories for storing hot code segments within an application. Static placement techniques focus on placing the most frequently executed portions of programs ...

19 Memory access patterns of parallel scientific programs
F. Darema-Rogers, G. F. Pfister, K. So
May 1987 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1987 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '87**, Volume 15 Issue 1
Publisher: ACM Press
Additional Information:

Full text available: pdf(1.24 MB) full citation, abstract, references, citings, index terms

A parallel simulator, PSIMUL, has been used to collect information on the memory access patterns and synchronization overheads of several scientific applications. The parallel simulation method we use is very efficient and it allows us to simulate execution of an entire application program, amounting to hundreds of millions of instructions. We present our measurements on the memory access characteristics of these applications; particularly our observations on shared and private data, their ...

20 Selecting Software Phase Markers with Code Structure Analysis

Jeremy Lau, Erez Perelman, Brad Calder

March 2006 **Proceedings of the International Symposium on Code Generation and Optimization CGO '06**

Publisher: IEEE Computer Society

Full text available: Publisher Site Additional Information: full citation, abstract, citings, index terms

Most programs are repetitive, where similar behavior can be seen at different execution times. Algorithms have been proposed that automatically group similar portions of a program's execution into phases, where samples of execution in the same phase have homogeneous behavior and similar resource requirements. In this paper, we present an automated profiling approach to identify code locations whose executions correlate with phase changes. These "software phase markers" can be used to easily dete ...

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+abstract:instruction +abstract:port, +abstract:data +abstrac

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: **instruction port data port cache** Found **28** of **215,186**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 28 of 28 Result page: previous 1 **2**

Relevance scale

21 Understanding the effects of wrong-path memory references on processor performance

Onur Mutlu, Hyesoon Kim, David N. Armstrong, Yale N. Patt

June 2004 **Proceedings of the 3rd workshop on Memory performance issues: in conjunction with the 31st international symposium on computer architecture WMPI '04**

Publisher: ACM Press

Full text available: pdf(378.83 KB) Additional Information: full citation, abstract, references, citings, index terms

High-performance out-of-order processors spend a significant portion of their execution time on the incorrect program path even though they employ aggressive branch prediction algorithms. Although memory references generated on the wrong path do not change the architectural state of the processor, they can affect the arrangement of data in the memory hierarchy. This paper examines the effects of wrong-path memory references on processor performance. It is shown that these references significantl ...

Keywords: cache pollution, data prefetching, execution-driven simulation, processor performance analysis, speculative execution, wrong path modeling, wrong-path memory references

22 Exploring configurations of functional units in an out-of-order superscalar processor

Stéphan Jourdan, Pascal Sainrat, Daniel Litaize

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture ISCA '95**, Volume 23 Issue 2

Publisher: ACM

Full text available: pdf(893.57 KB) Additional Information: full citation, abstract, references, cited by, index terms

This study has been carried out in order to determine cost-effective configurations of functional units for multiple-issue out-of-order superscalar processors. The trace-driven simulations were performed on the six integer and the fourteen floating-point programs from the SPEC 92 suite. We first evaluate the number of instructions allowed to be concurrently processed by the execution stages of the pipeline. We then apply some restrictions on the execution issue of different instruction classes i ...

23 Register file and memory system design: Dynamic addressing memory arrays with

physical locality
Steven Hsu, Shih-Lien Lu, Shih-Chang Lai, Ram Krishnamurthy, Konrad Lai
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture MICRO 35**
Publisher: IEEE Computer Society
Full text available: pdf(907.32 KB) Publisher Site
Additional Information: full citation, abstract, references, index terms

As pipeline width and depth grow to improve performance, memory arrays in microprocessors are growing in entries and ports. Arrays will increase in physical size, which prolongs the access time due to wiring delay. In order to boost clock frequency, these memory arrays must take multiple cycles to complete an access. This delays the scheduling of dependent instructions and affects overall performance. This paper proposes a different circuit organization to enable fast and slow accesses solely de ...

24 A low power unified cache architecture providing power and performance flexibility (poster session)
Afzal Malik, Bill Moyer, Dan Cermak
August 2000 **Proceedings of the 2000 international symposium on Low power electronics and design ISLPED '00**
Publisher: ACM Press
Full text available: pdf(249.19 KB) Additional Information: full citation, abstract, references, citings, index terms

Advances in technology have allowed portable electronic devices to become smaller and more complex, placing stringent power and performance requirements on the devices components. The M7CORE M3 architecture was developed specifically for these embedded applications. To address the growing need for longer battery life and higher performance, an 8-Kbyte, 4-way set-associative, unified (instruction and data) cache with programmable features was added to the M3 core. These features allow the a ...

25 Low power memory: Synonymous address compaction for energy reduction in data TLB
Chinnakrishnan S. Ballapuram, Hsien-Hsin S. Lee, Milos Prvulovic
August 2005 **Proceedings of the 2005 international symposium on Low power electronics and design ISLPED '05**
Publisher: ACM Press
Full text available: pdf(194.54 KB) Additional Information: full citation, abstract, references, index terms

Modern processors can issue and execute multiple instructions per cycle, often performing multiple memory operations simultaneously. To reduce stalls due to resource conflicts, most processors employ multi-ported L1 caches and TLBs to enable concurrent memory accesses. In this paper, we observe that data TLB lookups within a cycle and across consecutive cycles are often *synonymous* --- they go to the same page. To exploit this finding, we propose two new mechanisms --- *intra-cycle compa ...*

***Keywords**: low-power TLB, multi-porting, spatial and temporal locality*

26 Conjoined-Core Chip Multiprocessing
Rakesh Kumar, Norman P. Jouppi, Dean M. Tullsen
December 2004 **Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture MICRO 37**
Publisher: IEEE Computer Society
Full text available: pdf(369.99 KB) Additional Information: full citation, abstract, citings

Chip Multiprocessors (CMP) and Simultaneous Multi-threading (SMT) are two approaches

that have been proposed to increase processor efficiency. We believe these two approaches are two extremes of a viable spectrum. Between these two extremes, there exists a range of possible architectures, sharing varying degrees of hardware between processors or threads. This paper proposes conjoined-core chip multiprocessing - topologically feasible resource sharing between adjacent cores of a chip multiprocess ...

27 Power optimization of variable voltage core-based systems

Inki Hong, Darko Kirovski, Gang Qu, Miodrag Potkonjak, Mani B. Srivastava
May 1998 **Proceedings of the 35th annual conference on Design automation DAC '98**
Publisher: ACM Press

Full text available: pdf(435.37 KB) Additional Information: full citation, abstract, references, citings, index terms

The growing class of portable systems, such as personal computing and communication devices, has resulted in a new set of system design requirements, mainly characterized by dominant importance of power minimization and design reuse. We develop the design methodology for the low power core-based real-time system-on-chip based on dynamically variable voltage hardware. The key challenge is to develop effective scheduling techniques that treat voltage as a variable to be determined, in additio ...

Keywords: emulation, functional simulation, reconstruction, visibility

28 Memory optimization: A post-compiler approach to scratchpad mapping of code

Federico Angiolini, Francesco Menichelli, Alberto Ferrero, Luca Benini, Mauro Olivieri
September 2004 **Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems CASES '04**
Publisher: ACM Press

Full text available: pdf(154.30 KB) Additional Information: full citation, abstract, references, citings, index terms

ScratchPad Memories (SPMs) are commonly used in embedded systems because they are more energy-efficient than caches and enable tighter application control on the memory hierarchy. Optimally mapping code and data to SPMs is, however, still a challenge. This paper proposes an optimal scratchpad mapping approach for code segments, which has the distinctive characteristic of working directly on application binaries, thus requiring no access to either the compiler or the application source code - a c ...

Keywords: design automation, dynamic programming, embedded design, executable patching, memory hierarchy, optimization algorithm, post-compiler processing, power saving, scratchpad memory

Results 21 - 28 of 28 Result page: previous 1 **2**

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

OPTION 1
Enter keywords or phrases, select fields, and select operators Help

data port in Full Text & All Fields

AND instruction port in Full Text & All Fields

AND cache in Full Text & All Fields

» Note: If you use all three search boxes, the entries in the first two boxes take precedence over the entry in the third box.

OPTION 2
Enter keywords, phrases, or a Boolean expression Help

» Note: You may use the search operators <and> or <or> without the start and end brackets <>.

» Learn more about Field Codes, Search Examples, and Search Operators

» **Publications**

Select publications

- IEEE Periodicals
- IET Periodicals
- IEEE Conference F
- IET Conference Pr
- IEEE Standards

» **Other Resources** (Availab

- IEEE Books
- Educational Courses

» **Standard Status**
(Applies to IEEE Standards

Status All

» **Select date range**

Search latest content u

From year All

to Present

» **Display Format**

Citation Citatio

» **Organize results**

Maximum 100

Display 25 resu

Sort by Relevance

In Descending

Help Contact Us

© Copyright 20

Indexed by Inspec

Results for **"(data port<and>instruction port)<and>cache"**
Your search matched **7** of **1689216** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

New [Beta] Application Notes Powered by GLOBALSPEC

Modify Search

(data port<and>instruction port)<and>cache Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Search Options

View Session History

New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

IEEE/IET Books Educational Courses A

IEEE/IET Journals, transactions, letters, magazines, conference proceedings, and

view selected items Select All Deselect All

1. **Simulating Multimedia Systems with MVPSIM**
Jihong Kim; Yongmin Kim;
Design & Test of Computers, IEEE
Volume 12, Issue 4, Wiinter 1995 Page(s):18
Digital Object Identifier 10.1109/MDT.1995.473309
AbstractPlus | Full Text: PDF(816 KB) IEEE JNL
Rights and Permissions

2. **Distributed crossbar architecture for area-efficient combined data/instru multiple ports**
Johguchi, K.; Zhu, Z.; Hirakawa, T.; Koide, T.; Hironaka, T.; Mattausch, H.J.;
Electronics Letters
Volume 40, Issue 3, 5 Feb. 2004 Page(s):160 - 162
Digital Object Identifier 10.1049/el:20040130
AbstractPlus | Full Text: PDF(198 KB) IET JNL

3. **Single-chip H.324 videoconferencing**
Golston, J.;
Micro, IEEE
Volume 16, Issue 4, Aug. 1996 Page(s):21 - 33
Digital Object Identifier 10.1109/40.526922
AbstractPlus | References | Full Text: PDF(1148 KB) IEEE JNL
Rights and Permissions

4. **Simulating multimedia systems with MVPSIM**
Jihong Kim; Yongmin Kim;
Design & Test of Computers, IEEE
Volume 12, Issue 4, Winter 1995 Page(s):18 - 27
Digital Object Identifier 10.1109/54.491234
AbstractPlus | References | Full Text: PDF(3048 KB) IEEE JNL
Rights and Permissions

5. **Handling different computational granularity by a reconfigurable IC feat**

and a network-on-chip
Lertora, F.; Borgatti, M.;
Field-Programmable Custom Computing Machines, 2005. FCCM 2005. 13th A
18-20 April 2005 Page(s):45 - 54
Digital Object Identifier 10.1109/FCCM.2005.39

AbstractPlus | Full Text: PDF(192 KB) IEEE CNF
Rights and Permissions

6. **A 333-MHz dual-MAC DSP architecture for next-generation wireless appl**
Kolagotla, R.K.; Fridman, J.; Hoffman, M.M.; Anderson, W.C.; Aldrich, B.C.; W
Dunton, R.R.; Booth, L.A., Jr.;
Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01).
Conference on
Volume 2, 7-11 May 2001 Page(s):1013 - 1016 vol.2
Digital Object Identifier 10.1109/ICASSP.2001.941089

AbstractPlus | Full Text: PDF(328 KB) IEEE CNF
Rights and Permissions

7. **AMULET3: a high-performance self-timed ARM microprocessor**
Furber, S.B.; Garside, J.D.; Gilbert, D.A.;
Computer Design: VLSI in Computers and Processors, 1998. ICCD '98. Proce
Conference on
5-7 Oct. 1998 Page(s):247 - 252
Digital Object Identifier 10.1109/ICCD.1998.727058

AbstractPlus | Full Text: PDF(32 KB) IEEE CNF
Rights and Permissions

Help Contact Us

Indexed by Inspec

© Copyright 20